A New Channel Selection Algorithm for the Weightless-N Frequency Hopping with Lower Collision Probability

Riyadh A. Abbas, Ali Al-Sherbaz, Abdeldjalil Bennecer, Phil Picton
School of Art, Science and Technology
University of Northampton
Northampton, UK
{riyadh.abbas, Ali.Al-Sherbaz, Abdeldjalil.Bennecer, Phil.Picton }@northampton.ac.uk

***Abstract*— There are different techniques used by Machine-to-machine (M2M) communications technologies to mitigate collision problem and data loss. One of these techniques is Frequency Hopping (FH), which is used by Weightless-N technology with a special random channel selection algorithm. In such a system, the probability of a message collision mainly depends on the randomisation algorithm used to access channels. This paper provides a novel randomisation algorithm for the channel selection process of the Weightless-N system. The new proposed algorithm is based on a uniform randomisation distribution and called a Uniform Randomisation Channel Selection Technique (URCST). This new algorithm provides a better system performance and lower probability of collision. In addition, it is faster and easier than the Mersenne Twister algorithm.**

***Keywords*— Weightless-N; M2M; LPWAN; frequency hopping; random channel access; collision.**

I. Introduction

The development of Machine-to-Machine communications systems has been increasing recently, especially considering that they have a wide range of applications in smart cities and the Internet of Things (IoT). However, with the massive number of connected devices in such applications, the problem of message collision becomes a vital factor that significantly affects the reliability and performance of the M2M systems [1], [2].

Most M2M technologies focus on reducing the complexity and cost of the terminal devices to achieve the low cost and low power consumption essential requirements of the modern M2M communications system [3]–[5]. Therefore, the majority of these technologies eliminate the synchronisation process and rely on the random channel access techniques. For instance, Dash7 and OnRamp use the random time slotted Aloha method [2], [6]. Sigfox uses the Listen Before Talk (LBT) Adaptive Frequency Agility (AFA) technique while LoRa utilises the LBT technique with a random time access technique [7]. On the other hand, Weightless-N employs a frequency hopping technique with a random channel selection algorithm, which is based on the identification number (ID) and the internal timer of the terminal devices.

The next part of the paper will explain the characteristics of the Weightless-N technology and the standard channel selection algorithm that is used in Europe only. In Section-III, the paper clarifies the new channel selection algorithm (URCST) with a comparison to the Mersenne Twister method. In Section-IV, the paper explains the method used to evaluate and test the system for both the standard and the URCST algorithms. It also provides a comparison of the system performance with the standard algorithm, the URCST algorithm, and the MT algorithm in terms of the total number of lost messages.

II. Weightless-N Technology

In April 2015, the Weightless Special Interest Group (SIG) published the final release of the Weightless-N standard version 1.0. The Weightless-N technology is a star M2M network system where all terminal devices communicate with a central base station. It is based on a uni-directional uplink-only system solution in which all terminals send messages to the central base station without synchronisation or acknowledgement [8]. With such system architecture, there is a high probability of collisions and lost messages. To mitigate the collision problem, Weightless-N proposes a special frequency hopping scheme that randomises the selected channel for each transmission.

The following sections explain the general characteristics of the Weightless-N technology and the frequency hopping algorithm.

A. *General characteristics of Weightless-N*

Weightless-N utilises the sub-GHz ISM band of 868 MHz in Europe with the differential binary phase shift keying (DBPSK) modulation scheme. Furthermore, Weightless-N divides this band into six wide bands, as shown in Table 1 [8]. Each base station will be associated with one of these six wide bands and can detect all transmissions within its range of frequency. On the other hand, terminal devices work on a narrow frequency band of 200 Hz called micro-channels. This offers a large number of narrowband channels that can be used by any terminal.

This research was sponsored by Wasit University, Ministry of Higher Education, Iraq.

Furthermore, Weightless-N divides each wideband into three sub-bands called macro-channels, each contains a number of micro-channels. For example, each macro-channel in the 0.6 MHz band contains 1000 channels.

TABLE 1
WEIGHTLESS-N FREQUENCY BANDS

Band No.	***Lower Band (MHz)***	***Upper Band (MHz)***	***Bandwidth (MHz)***	***Number of Channels***
1	863	864.998	1.998	9990
2	865	868	3	15000
3	868	868.6	0.6	3000
4	868.7	869.2	0.5	2499
5	869.4	869.64	0.24	1200
6	869.7	870	0.3	1500

To increase the chance of receiving the message correctly, each terminal sends three identical copies of each message on different macro-channels and micro-channels using the frequency hopping regime, as shown in Fig. 1. However, according to the Weightless-N standard, the total number of message copies can be increased up to 8 for applications requiring a high QoS.

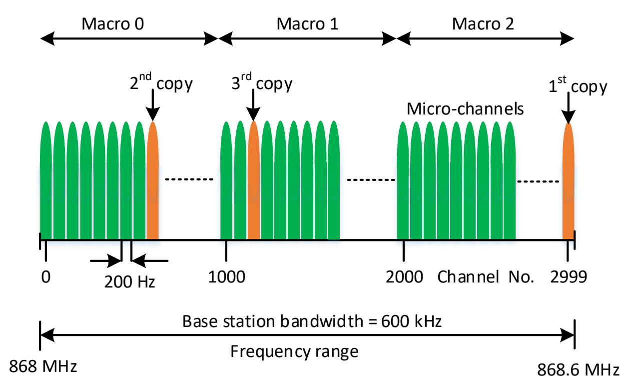

Fig. 1. Weightless-N channels

Each message sent from a terminal consists of 7 blocks as shown in Table 2, where FCS represents a Frame-Check Sequence to indicate any error in the message. The base station will check the FCS and if any error occurs, the message will be neglected. Otherwise, the base station will check the timestamp, which is a count of minutes on the terminal's internal timer. Messages that have the same timestamp are assumed to be copies of the same message. Therefore, the maximum message rate of Weightless-N is one message per minute with a data rate of 100 bps [8].

TABLE 2
THE WEIGHTLESS-N MESSAGE

Preamble	***ID***	***Data length***	***Time stamp***	***Payload***	***MAC***	***FCS***
3 bytes	6 bytes	5 bits	19 bits	0 – 20 bytes	24 bits	16 bits

B. Frequency hopping algorithm

The channel selection algorithm is based on the two least significant bytes of both the ID and the internal timer of the terminal counting in seconds. Assuming that the two least significant bytes of the ID is represented by *0xZZZZ* and the two least significant bytes of the timer are *0xMMSS*, the channel selection would be as follows [8]:

1) Macro-channel selection:

The first macro-channel M1, which will be used to send the first message copy, will be selected according to the formula:

$$M1 = 0xSS \bmod 3 \quad (1)$$

The second and third hops will be selected according to the formula:

$$0xSS \bmod 2 \quad (2)$$

If the result of the second formula equals zero, then the lower remaining channel index will be chosen for the second hop. Otherwise, the highest index will be used.

2) Micro-channel selection:

Assuming that the total number of channels in each macro-channel is NC, the micro-channel's number (mc) for each hop will be given by the formulas:

$$mc1 = (0xZZZZ \;\; XOR \;\; 0xMMSS) \bmod NC \quad (3)$$
$$mc1 = (0xZZZZ \;\; OR \;\; 0xMMSS) \bmod NC \quad (4)$$
$$mc1 = (0xZZZZ \;\; AND \;\; 0xMMSS) \bmod NC \quad (5)$$

For message copies that are more than three, the same procedure will be used to define the next hop. However, formulas 4 and 5 provide a high probability of selecting the same channel from different terminals at the same time.

III. THE URCST ALGORITHM

The URCST algorithm employs the same macro-channel selection process used by the standard algorithm, which was described by formulas 1 and 2. The procedure will be repeated for any message copy larger than three. On the other hand, the micro-channel selection procedure utilises a random number generator that is based on a ring shift register of the internal timer of the terminal. For each hop, the micro-channel will be selected by shifting the timer *0xMMSS* to the left by one bit and the most significant bit will be fed to the least significant bit of the register. The resulting number of the ring register will be *XOR* with the terminal's *ID*, as shown in Fig. 2. This reduces the probability of selecting the same channel by different terminals at the same time and provides a better channel distribution in comparison with the standard algorithm.

If *n* represent the number of message copies, the micro-channel number for the current hop can be obtained by the formula:

$$mc_i = (0xZZZZ \;\; XOR \;\; (0xMMSS << i)) \bmod NC \quad (3)$$

$$\text{where } i = 0,1,2,\ldots n\text{-}1$$

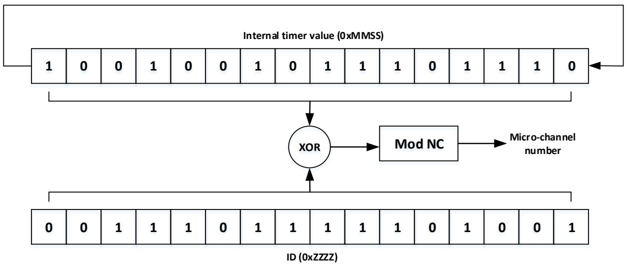

Fig. 2. URCST algorithm channel selection using the cyclic shift register.

Fig. 3 gives the pseudocode for the URCST algorithm, which was implemented using the MATLAB software.

URCST Algorithm: Pseudocode of channel selection algorithm

```
SET t = Timer, id = ID, n = number of message copies, M =
Macros, mc = Micros, NC = number of channels
T1 = t AND 0xFF
M(1) = T1 mod 3
If (T1 mod 2) == 0
   M(2,3) = sort remaining values of 1,2,3 ascendingly
Else
   M(2,3) = sort remaining values of 1,2,3 in a descending
order
End
T2 = t AND 0xFFFF
id = id AND 0xFFFF
For i = 0 to n-1
   Select Macro from M
   mc(i) = (T2 XOR id) mod NC
   B15 = T2 AND 32768
   B15 = B15 >> 15
   T2 = T2 << i+1
   T2 = T2 OR B15
end
```

Fig. 3. Pseudocode of the URCST algorithm.

IV. System Evaluation and Performance Analysis

The evaluation of the system was implemented using the MATLAB software. it considers only the interference between devices that are connected to the same base station, assuming ideal channel and neglecting any other source of interference. The test employs the 0.6 MHz band with 3000 channels. To achieve more realistic results, the total number of devices were divided into four groups with different characteristics as follows:

- *Group 1:* represents 40% of the total number of devices and each device send a message randomly in the period of 1 – 2 minutes. Each message payload was set to be 4 bytes.
- *Group 2:* represents 20% of the total number of devices and each device send a message periodically each 1 minute. Each message payload was set to be 6 bytes.
- *Group 3:* represents 20% of the total number of devices and each device send a message randomly in the period of 2 – 5 minutes. Each message payload was set to be 10 bytes.
- *Group 4:* represents 20% of the total number of devices and each device send a message periodically each 5 minutes. Each message payload was set to be 12 bytes.

The simulation was carried out for one hour assuming that all devices start working randomly within the first 10 minutes. The percentage of the lost messages shown on the graphs represent the total number of the lost messages from all devices for one hour.

The analysis shows a comparison between the standard algorithm and the URCST algorithm in terms of channel histogram and percentage of lost messages. In addition, the analysis shows the probability of collisions in the case of using the Mersenne Twister random number generator (MT19937) with the device's ID as a seed number to select a different random micro- channel on each hop. The results show a better system performance with lower collision probability using the URCST algorithm. It also provides almost the same system performance in comparison with the Twister algorithm. However, implementing Twister algorithm using microcontrollers has many disadvantages. First, the Twister algorithm requires a 32-bit microcontroller with a relatively high memory capacity. Second, the complexity of MT19937 is moderately high since a high number of iterations and two matrix multiplications are required to generate each random number. For instance, it requires $(n\times 624)$ loops to transmit the n^{th} message copy while the URCST algorithm needs only n loops. The complexity of the MT19937 algorithms can be expressed, according to the Big-O notation, as $O(n\times 624)$, while the URCST algorithm complexity is $O(n)$, where $n=1,2,\ldots 8$. Finally, MT19937 consumes much higher power which is one of the most important factors that restrict the use of this algorithm in M2M systems [9]–[12].

A. *Channel histogram*

The channel histogram represents the total number of sent messages on each micro-channel and the total number of lost messages on these channels. The analysis has been made with a total number of 4000 terminal devices and $n = 3$. The standard algorithm channel histogram shown in Fig. 4 shows that the channel request is crowded at the first part of each macro-channel. This gives a non-uniform channel distribution and increases the probability of collisions and leads to a high number of lost messages.

On the other hand, the URCST algorithm provides a much better uniform channel distribution among all micro-channels as shown in Fig 5. This significantly reduces the probability of collision and the total number of lost messages.

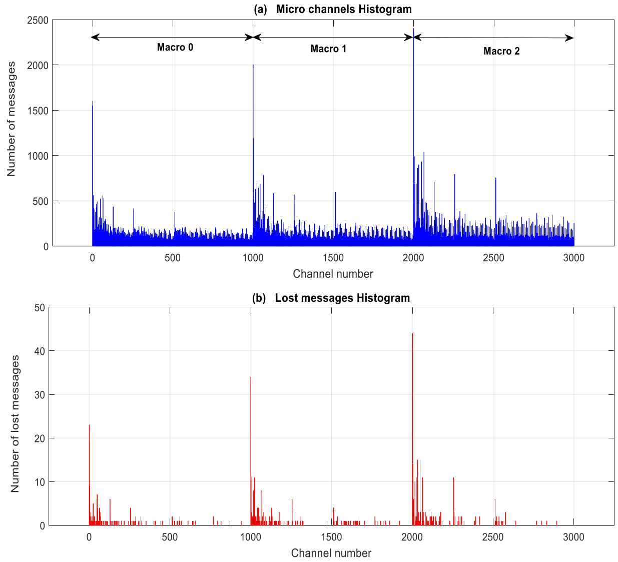

Fig. 4. Channel histogram of the standard algorithm with a total number of 4000 devices and $n=3$. (a) The total number of sent messages = 128237. (b) The total number of lost messages = 690 = 0.54 % of total messages.

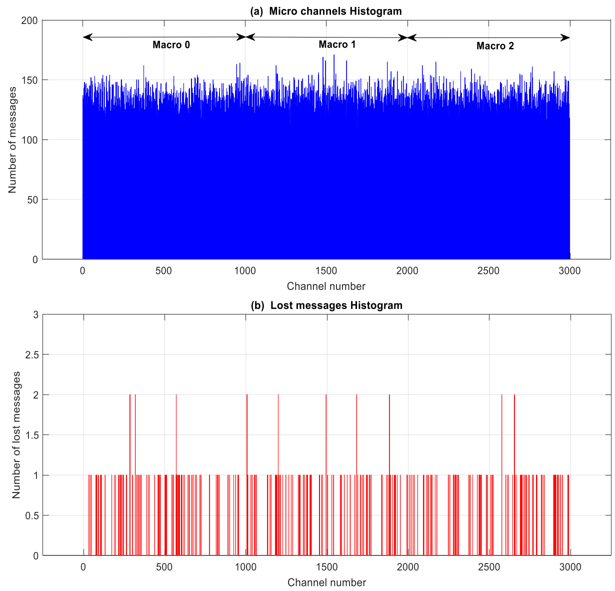

Fig. 5. Channel histogram of the URCST algorithm with a total number of 4000 devices and $n=3$. (a) The total number of sent messages = 128189. (b) The total number of lost messages = 230 = 0.18 % of total messages.

B. *Variable number of message copies*

Fig. 6 shows the percentage of the total number of lost messages for the standard algorithm, the URCST algorithm, and the MT19937 versus different number of message copies. Both the URCST algorithm and the MT19937 provide nearly the same percentage of lost messages that declines as the number of message copies increase. On the other hand, the standard algorithm provides the best performance at four message copies. Nevertheless, the percentage of lost messages rises for more than four message copies.

C. *Variable number of devices*

As the number of connected devices increases, the probability of collision rises up. The analysis shows that the total number of lost messages rises exponentially as the number of devices increases. It also shows that as the number of devices increases the URCST algorithm provides better performance. Fig. 7 shows the percentage of lost messages versus a different number of devices for $n = 3$.

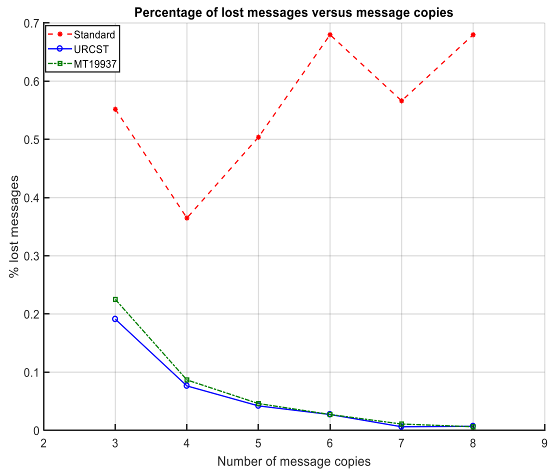

Fig. 6. The percentage of lost messages versus the number of message copies n with a total number of 4000 devices.

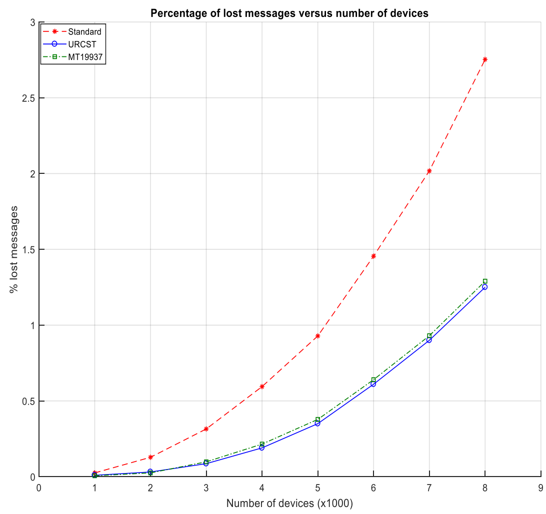

Fig. 7. The percentage of lost messages versus the number of terminal devices for a $n=3$.

V. CONCLUSION

As shown by the analysis of the system with three different algorithms, the URCST algorithm provides a better performance with a lower probability of collisions in comparison to the standard algorithm. Moreover, the URCST algorithm provides a better performance as the number of message copies increase while the standard algorithm provides the best performance at four message copies. This might be very useful for applications requiring a high QoS like security, fire alarms, heart disease monitoring, and Electronic Point of Sale (EPOS) [13], [14]. On the other hand, the URCST algorithm can be implemented using a simple microcontroller with much less complexity, computational time, and power consumption in comparison to the MT19937 algorithm.

For future work, a mathematical model will be derived for the probability of collision using the URCST algorithm for the channel selection process in Weightless-N technology. Since the URCST provides a uniform distribution over all channels and over different time specifications, it is fairly acceptable to assume that the channel selection process is independent of the time domain. This can significantly reduce the modelling complexity due to the separation of time domain and frequency domain. In such a case, the final probability of collision can be obtained by using the probability rules for independent events.

In addition, the URCST algorithm could be implemented in other technologies that are based on hopping technique. For instant, EnOcean technology employs the time hopping mechanism over 40 time slots with three message copies that are called sub-telegrams. The URCST algorithm will be implemented for time slot selection for EnOcean technology and evaluate the system performance in terms of the probability of collision.

ACKNOWLEDGMENT

This research was funded by the Ministry of Higher Education and scientific research in the Republic of Iraq, according to the scholarship number (4593) in (11/11/2014) to sponsor the first author to pursue his PhD research.

REFERENCES

[1] C. Anton-Haro and M. Dohler, "Introduction to machine-to-machine (M2M) communications," in *Machine-to-machine (M2M) Communications Architecture, Performance and Applications*, C. Anton-Haro and M. Dohler, Eds. Cambridge: Woodhead Publishing, 2015, pp. 27–46.

[2] X. Vilajosana, P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Standardized Low-Power Wireless Communication Technologies for Distributed Sensing Applications," *Sensors*, vol. 14, no. 2, pp. 2663–2682, 2014.

[3] W. Webb, "Weightless machine-to-machine (M2M) wireless technology using TV white space," in *Machine-to-machine (M2M) Communications Architecture, Performance and Applications*, C. Anton-Haro and M. Dohler, Eds. Cambridge: Elsevier, 2015, pp. 93–108.

[4] A. Laya, K. Wang, L. Alonso, J. Alonso-Zarate, and J. Markendahi, "Supporting machine-to-machine communications in long-term evolution networks," in *Machine-to-machine (M2M) Communications Architecture, Performance and Applications*, C. Anton-Haro and M. Dohler, Eds. Cambridge: Woodhead Publishing, 2015, pp. 109–129.

[5] W. Webb, "Weightless: The Technology to Finally Realise the M2M Vision," *Int. J. Interdiscip. Telecommun. Netw.*, vol. 4, no. 2, pp. 30–37, 2012.

[6] M. Arsalan, A. Umair, and V. K. Verma, "Dash7 : Performance," *J. Electron. Commun. Eng.*, vol. 2, no. 5, pp. 8–11, 2012.

[7] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios," in *IEEE Wireless Communications*, 2015, pp. 1–7.

[8] Weightless-SIG, "Weightless-N System specification v1.0," Cambridge, 2015.

[9] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[10] P. Echeverría and M. López-Vallejo, "High performance FPGA-oriented mersenne twister uniform random number generator," *J. Signal Process. Syst.*, vol. 71, no. 2, pp. 105–109, 2013.

[11] M. Saito and M. Matsumoto, "A Variant of Mersenne Twister Suitable for Graphic Processors," *Trans. Math. Softw.*, vol. 39, no. 2, pp. 1–20, 2013.

[12] V. Bonato, B. F. Mazzotti, M. M. Fernandes, and E. Marques, "A mersenne twister hardware implementation for the Monte Carlo localization algorithm," *J. Signal Process. Syst.*, vol. 70, no. 1, pp. 75–85, 2013.

[13] Ofcom, "M2M application characteristics and their implications for spectrum Final Report," London, 2014.

[14] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, Jan. 2015.